1. Remove Duplicates from a String?

Ans –

public class Main {

public static void main(String[] args) {

String inputString = "examplestringwithduplicates";

StringBuilder uniqueString = new StringBuilder();

for (int i = 0; i < inputString.length(); i++) {

char currentChar = inputString.charAt(i);

if (uniqueString.indexOf(String.valueOf(currentChar)) == -1) {

uniqueString.append(currentChar);

}

}

System.out.println("String after removing duplicates: " + uniqueString.toString());

}

}

2. Print Duplicates Characters from a String?

Ans –

public class Main {

public static void main(String[] args) {

String inputString = "examplestringwithduplicates";

for (int i = 0; i < inputString.length(); i++) {

char currentChar = inputString.charAt(i);

boolean isDuplicate = false;

for (int j = i + 1; j < inputString.length(); j++) {

if (currentChar == inputString.charAt(j)) {

isDuplicate = true;

break;

}

}

if (isDuplicate) {

System.out.print(currentChar + " ");

}

}

}

}

3. Check if "2552" is Palindrome?

Ans - public class Main {

public static void main(String[] args) {

String inputString = "2552";

boolean isPalindrome = true;

for (int i = 0, j = inputString.length() - 1; i < j; i++, j--) {

if (inputString.charAt(i) != inputString.charAt(j)) {

isPalindrome = false;

break;

}

}

if (isPalindrome) {

System.out.println(inputString + " is a palindrome.");

} else {

System.out.println(inputString + " is not a palindrome.");

}

}

}

4. Count Consonants, Vowels, and Special Characters in a String?

Ans - public class Main {

public static void main(String[] args) {

String inputString = "example string with 123 special characters!";

int vowels = 0;

int consonants = 0;

int specialCharacters = 0;

for (char ch : inputString.toLowerCase().toCharArray()) {

if (ch >= 'a' && ch <= 'z') {

if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {

vowels++;

} else {

consonants++;

}

} else {

specialCharacters++;

}

}

System.out.println("Vowels: " + vowels);

System.out.println("Consonants: " + consonants);

System.out.println("Special Characters: " + specialCharacters);

}

}

5. Anagram Checking (Least Inbuilt Methods)?

Ans –

public class Main {

public static void main(String[] args) {

String str1 = "listen";

String str2 = "silent";

if (areAnagrams(str1, str2)) {

System.out.println(str1 + " and " + str2 + " are anagrams.");

} else {

System.out.println(str1 + " and " + str2 + " are not anagrams.");

}

}

private static boolean areAnagrams(String str1, String str2) {

if (str1.length() != str2.length()) {

return false;

}

int[] count = new int[256];

for (int i = 0; i < str1.length(); i++) {

count[str1.charAt(i)]++;

count[str2.charAt(i)]--;

}

for (int value : count) {

if (value != 0) {

return false;

}

}

return true;

}

}

6. Pangram Checking (Least Inbuilt Methods)?

Ans - public class Main {

public static void main(String[] args) {

String sentence = "The quick brown fox jumps over the lazy dog";

if (isPangram(sentence.toLowerCase())) {

System.out.println("The sentence is a pangram.");

} else {

System.out.println("The sentence is not a pangram.");

}

}

private static boolean isPangram(String sentence) {

boolean[] alphabetPresent = new boolean[26];

for (int i = 0; i < sentence.length(); i++) {

char ch = sentence.charAt(i);

if (ch >= 'a' && ch <= 'z') {

alphabetPresent[ch - 'a'] = true;

}

}

for (boolean isPresent : alphabetPresent) {

if (!isPresent) {

return false;

}

}

return true;

}

}

7. Unique Characters in a String?

Ans –

public class Main {

public static void main(String[] args) {

String inputString = "uniquecharacters";

if (hasUniqueCharacters(inputString)) {

System.out.println("The string has all unique characters.");

} else {

System.out.println("The string does not have all unique characters.");

}

}

private static boolean hasUniqueCharacters(String inputString) {

for (int i = 0; i < inputString.length(); i++) {

for (int j = i + 1; j < inputString.length(); j++) {

if (inputString.charAt(i) == inputString.charAt(j)) {

return false;

}

}

}

return true;

}

}

8. Maximum Occurring Character in a String?

Ans –

public class Main {

public static void main(String[] args) {

String inputString = "examplestring";

char maxChar = '\0';

int maxCount = 0;

for (int i = 0; i < inputString.length(); i++) {

char currentChar = inputString.charAt(i);

int currentCount = 1;

for (int j = i + 1; j < inputString.length(); j++) {

if (inputString.charAt(j) == currentChar) {

currentCount++;

}

}

if (currentCount > maxCount) {

maxCount = currentCount;

maxChar = currentChar;

}

}

System.out.println("Maximum occurring character: " + maxChar);

}

}